

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КОНОТОПСЬКИЙ ІНСТИТУТ

Кафедра електронних
приладів і автоматики

Кваліфікаційна робота бакалавра
Акустично-вібраційний перетворювач для оцінки розташування об'єктів

студента гр. ЕІ-61к
Науковий керівник,
викладач

Р.В. Худяков

М.В.Бібик

Конотоп 2020

РЕФЕРАТ

Об'єктом дослідження виступає процес роботи акустично-вібраційного перетворювача.

Метою дослідження є проектування акустично-вібраційного перетворювача для оцінки розташування об'єктів.

Теоретичні дослідження роботи зосереджені на допомозі орієнтування незрячих людей на місцевості. Основна мета теоретичного дослідження полягає у розробленні даного приладу на основі даних літературних джерел тематики дослідження.

У результаті проведених досліджень встановлено, що акустично-вібраційний перетворювач має просту складову та може допомогти людині з орієнтуванням на місцевості.

Робота викладена на 30 сторінках, у тому числі включає 13 рисунків, список цитованої літератури із 13 джерел.

КЛЮЧОВІ СЛОВА: АКУСТИЧНО-ВІБРАЦІЙНИЙ ПЕРЕТВОРЮВАЧ, РОЗТАШУВАННЯ ОБ'ЄКТІВ, ARDUINO .

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	5
1.1 Особливості орієнтування в просторі у сліпих.....	5
1.2 Засоби орієнтації у просторі людей з обмеженими можливостями (вадами зору)	9
РОЗДІЛ 2 ПОСТАНОВКА ЗАДАЧІ	16
2.1 Мета та задачі.....	16
2.2 Вибір засобів реалізації.....	16
РОЗДІЛ 3 ПРОЕКТУВАННЯ АКУСТИЧНО-ВІБРАЦІЙНОГО ПЕРЕТВОРЮВАЧА.....	18
3.1 Структура проекту.....	18
3.2 Проектування схеми	20
3.3 Програмна частина	25
ВИСНОВКИ.....	28
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	29
ДОДАТОК А Програмний код приладу	

ВСТУП

Враховуючи велику кількість людей із вадами зору, доцільно розробити акустично-вібраційний перетворювач для оцінки розташування об'єктів.

Використання акустично-вібраційного перетворювача допоможе людині з орієнтацією в суспільстві. При впровадженні доповненого алгоритму в архітектуру додатка для Android враховує сучасні тенденції повного впровадження інформаційних технологій у всі сфери повсякденного життя та професійного життя людини.

Об'єктом дослідження виступає процес роботи акустично-вібраційного перетворювача.

Предмет дослідження – робота приладу з можливістю оцінки розташування об'єктів.

Метою дослідження є проектування акустично-вібраційного перетворювача для оцінки розташування об'єктів.

Теоретичні дослідження роботи зосереджені на допомозі орієнтування незрячих людей на місцевості. Основна мета теоретичного дослідження полягає у розробленні даного приладу на основі даних літературних джерел тематики дослідження.

Експериментальні дослідження полягають у перевірці адекватності спроектованого приладу, шляхом тестування системи, використовуючи її для орієнтування в приміщенні тестувальників в електронному вигляді.

Практична значимість полягає у проектуванні акустично-вібраційного перетворювача для оцінки розташування об'єктів.

Використання розробленої системи дозволить незрячій людині орієнтуватися в навколишньому середовищі. [1]

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Особливості орієнтування в просторі у сліпих

Експериментальні матеріали вітчизняних вчених показали, що як зір, так і дотик спрямовані на відображення об'єктивних предметів і об'єктів навколишнього світу і що вони адекватно відтворюють однакові фізичні властивості предметів, але різними способами. Дослідження, присвячені особливостям сприйняття простору незрячими, свідчать і про те, що сліпі мають можливість миттєво сприймати об'єкти пізнання при системному використанні збережених аналізаторів.

Сприйняття навколишнього світу відбувається нерозривно з розвитком мови і мислення, за допомогою яких сліпі абстрагують і узагальнюють ознаки і властивості сприймаються предметів.

У прямій залежності від сприйняття об'єктів пізнання знаходяться подання, в яких відображаються раніше сприйняті властивості та ознаки предметів. Вивчення уявлень сліпих вітчизняними дослідниками (М. І. Земцова, В. А. Кручинін, Б. І. Коваленко, Ю. А. Кулагін, А. Г. Литваком, Е. Б. Островської, В. С. Сверлова, Л. І. Солнцевої, Ф. Н. Шемякін та ін.) свідчить про те, що «просторові уявлення незрячих хоча і мають деякі особливості, в цілому адекватно відображають зовнішній світ». Так, А. Г. Литвак пише: «Особливості уявлень сліпих свідчать, що їх образи пам'яті в більшій своїй частині менш точні, повні, узагальнені в порівнянні з уявленнями нормально бачать, що зорові і відчутні образи мають суттєві відмінності, однак уявлення, якими оперують сліпі в тому або іншому ступені, адекватно відображають дійсність ».

Здатність сліпого орієнтуватися в просторі знаходиться в прямій залежності від рівня сформованості у нього просторових уявлень. Тим часом,

створення у сліпих правильних образів - дуже важлива і важке завдання. Першорядне значення в діяльності сліпих мають сенсорно-перцептивні процеси, в структурі яких можна виділити дії по розрізнення, відбору, переробці інформації і створення на їх основі образу і опорного уявлення про досліджуваних явищах і предметах навколишньої дійсності.

Сенсорно-перцептивні процеси сліпого засновані на системній діяльності збережених аналізаторів (слухового, рухового та ін.), Кожен з яких вносить в цю систему свій специфічний внесок. Все це необхідно враховувати при роботі з просторової орієнтації сліпих. Розглядаючи особливості орієнтації сліпих ми назвемо кошти розвитку їх мобільності. Під мобільністю розуміють «безпечне та ефективне пересування в навколишньому середовищі». З цією метою використовуються орієнтири, що формують для сліпого термінову інформацію про навколишній простір. Орієнтири дозволяють локалізувати (виділити) себе і навколишні предмети для визначення свого місця розташування і організувати контроль над напрямком переміщення в просторі.

Уміння сприймати орієнтири визначає мобільність сліпого в навколишньому середовищі. Орієнтир - це показчик на що-небудь (місцезнаходження предмета, напрямок шляху і т. П.). Орієнтирами можуть бути сторони горизонту (північ, південь, захід, схід), запахи, звуки. Орієнтири можуть бути опорними, проміжними, контрольними. «До опорним орієнтирам відносяться природні і штучні об'єкти, добре відомі всім або більшості жителів даного міста або сільській місцевості.

При орієнтуванні люди зазвичай використовують опорні і проміжні орієнтири. Проміжні орієнтири допомагають вибрати і зберегти напрямок шляху. Вони можуть бути постійними і тимчасовими. Контрольні орієнтири дозволяють визначити своє становище на місцевості ».

Розрізняють зорові, слухові, нюхові орієнтири, т.д. В залежності від виду чутливості, на основі якої вони створюються. Однак у дітей з порушенням зору в процесі орієнтування в просторі не представляється

можливим здійснювати сприйняття необхідних орієнтирів без використання вербальних і технічних засобів. Вміле застосування спеціальних засобів при орієнтації в просторі значно впливає на формування мобільності сліпого.

До технічних засобів відносять орієнтування пристосування, наприклад, пристрій «звуковий маяк», створений І. Р. Тіновським і Н. В. Мошенніковим, який допомагає сліпим орієнтуватися при виконанні локомоторних дій (в ходьбі, бігу, стрибках, метаннях в мету і т. д.). Цікавим є застосування радіозв'язку, описане Л. А. Семеновим і Л. І. Солнцевої. Значення мають кошти «спрямованого відчуття рухів», які беруть участь у формуванні рухових здібностей дітей. Деяке поширення в практиці орієнтування отримав ультразвуковий локатор, виготовлений в Англії, - «Ультрасоннік».

На основі звукового сигналу, що змінюється за тембром і частоті, можна виявити такі перешкоди, як стіл, будова, паркан і т. П. Використовується в орієнтуванні і ультразвуковий локатор «Орієнтир», що нагадує ліхтарик. «Якщо вузький промінь ліхтарика потрапляє на перешкоду, то в мініатюрному головному телефоні виникає певний сигнал. По висоті тону можна судити про відстань до виявленого предмета, за тембром - про його характер, а зміна висоти сигналу попередить, що предмет рухається. Таким чином, можна оглядати місцевість в радіусі до десяти метрів ».

Технічні засоби орієнтування для сліпих ґрунтуються на створенні звукового або тактильного сигналу, що забезпечує формування інформації про власний пересуванні сліпого в просторі. До засобів орієнтування також відносяться фоноскопія.

Муратова, прилади В. Старковіч і Т. Кулішевського, що дають тактильні сигнали на шкіру чола. Однак особливе практичне значення у формуванні вмінь і навичок мобільності в просторі у сліпих мають тростини (звичайні, довгі, складні, лазерні). Тростини відрізняються по довжині, матеріалом, формою ручок і наконечників. Вони «розширюють зону обстеження,

допомагають сліпим виявляти різні орієнтири і перешкоди, тростини виконують функції щупа, буфера, генератора звуку».

Палиця є основним і одним з найбільш універсальних технічних засобів орієнтування. Далі розглянемо вербальні засоби наочності, які відіграють найважливішу роль в отриманні інформації про оточуючих сліпого предметах і об'єктах. В результаті мовного спілкування відбувається узгодження, уточнення, взаємне доповнення відображення дійсності.

У процесі самотійного пересування в просторі сліпий змушений звертатися до перехожих для підтвердження уявлення про своє становище на маршруті або місці того об'єкта, до якого він йде. У перехожих незрячий може дізнатися всю необхідну йому інформацію. Отже, роль засобів наочності при формуванні вмінь та навичок орієнтування та мобільності у сліпих учнів зводиться до отримання термінової інформації про навколишній простір, без якої неможливо самотійне пересування сліпого. Робота з розвитку просторових уявлень у сліпих спрямована на реалізацію наступних основних завдань:

- формувати вміння визначати просторове розташування предметів по відношенню до самого себе і по відношенню до інших об'єктів;
- розвивати уявлення про взаєморозташування об'єктів в просторі по вертикальній та горизонтальній вісях;
- розвивати вміння самотійно вживати прийменники, які позначають розташування об'єктів в просторі в промові. [2]

1.2 Засоби орієнтації у просторі людей з обмеженими можливостями (вадами зору)

1) Спеціальний iPad для людей з вадами зору

Корпорація iPad випустила планшет iPad для сліпих. Він має назву - iSense. Планшет буде відрізнятися від звичайного iPad тим, що на екрані планшета буде накладена спеціальна світлочутлива плівка (протектор). І над тими ділянками екрану, які дають яскравіше випромінювання, плівка стає більш опуклою. Так спеціальна технологія ловить світлові хвилі, що надходять з екрана, і перетворює їх на літери Брайля та піктограми, зрозумілі незрячим. Однак зараз у зв'язку зі смертю Джобса не ясно, чи буде реалізована ця концепція.



Рисунок 1.1 iPad для сліпих[3]

2) GPS-Навігатор для незрячих

На корпусі GPS-навігатора для слів є клавіатура шрифтом Брайля, а також навушники, мікрофон та бездротове підключення до Інтернету. Система постійно постачає сліпця інформацією про його місцезнаходження, про те, якою дорогою він йде, що за будівлі розташовані поруч.

Відбувається це так: сліпий каже в мікрофон про своїх координатах в даний момент і про те місце, куди він збирається. Система відповідає: «Повернутися на 15 градусів ліворуч і пройти 70 метрів по тротуару». Якщо при цьому сліпий сходить з тротуару, «робот» робить йому зауваження. А за допомогою інтернет-зв'язку сліпий може повідомити близьким, що він, наприклад, заблукав.



Рисунок 1.2 GPS-навігатор для сліпих[3]

Інший навігатор - шведський концепт гаджета Munivo зможе скласти собакам-поводирів серйозну конкуренцію. Апарат використовує для орієнтації в просторі мініатюрний ехолотатор, який «обмацує» простір і створює картинку в електронному мозку гаджета. Спираючись на ці дані, комп'ютер створює оптимальну для людини траєкторію руху і швидко

попереджає його про перешкоди: ліхтарних стовпах або йдуть назустріч людей.



Рисунок 1.3 GPS-навігатор для сліпих з роботою через телефон[3]

Наступний крок - інформування користувача про те, що відбувається. Для цього надолонник Munivo обладнаний спеціальними панелями з силікону. За допомогою чотирьох активних точок в різних комбінаціях апарат зможе рекомендувати слабозрячим користувачеві повернути праворуч, піти вперед або різко зупинитися.

3) Стільниковий телефон без дисплея

Корейські дизайнери Кім Янгс і Йом Юнсол розробили концепт мобільника для людей з вадами зору, який отримав назву Voim. У нього відсутня дисплей, замість нього - панель для генерації повідомлень в абетці Брайля в вигляді висуваються точок.

Модель також отримала фотокамеру, Bluetooth-гарнітуру, технології розпізнавання тексту і синтезу голосу. Так, сліпому досить буде провести камерою перед об'єктом, телефон проведе сканування і озвучить основні характерні риси: колір, форму, наявність тексту.



Рисунок 1.4 Стільниковий телефон без дисплея[3]

Крім того, телефон можна носити на грудях камерою вперед, так що дані про знаходяться попереду перешкоди можуть надходити безперервно.

4) Фотоапарат без екрану

Перша у світі цифрова камера для сліпих під назвою Touch Sight не має РК-екрана. Але є тактильний дисплей Брайля, який відображає зображення в рельєфі та крапці. Під час тестів з'ясувалося, що орієнтувати камеру найкраще, прикріпивши дисплей до лоба.



Рисунок 1.5 Фотоапарат без екрану Touch Sight[3]

5) Кредитна картка яка озвучує список покупок

Концепція кредитної картки, розроблена Кваном Кі Нам, працює таким чином, що "розпізнає" та ідентифікує користувача за відбитком пальців. Палець слід підняти до тієї частини кредитної картки, де вона зазвичай підписується власником картки. Новинка включає інтерактивний дисплей Брайля та аудіодинаміки, щоб відтворити, які речі були придбані.

6) Розумні окуляри "освітять наскрізь"

Вчені Оксфордського університету розробили "розумні окуляри», які оснащені мініатюрними камерами, світлодіодами і кишеньковим комп'ютером. Камери фіксують зображення, а потім передають інформацію комп'ютера. Мініатюрний пристрій обробляє дані, після чого посиляє сигнали власнику окулярів у вигляді світлових спалахів. За допомогою цих сигналів незряча людина може дізнатися, який об'єкт знаходиться перед ним.

У комплект окулярів також входять навушники. Камери зчитують інформацію, яка після обробки передається через навушники. Це допомагає розрізняти номери автобусів або перевіряти розклад поїздів.

Однак окуляри не можуть допомогти абсолютно сліпим людям, вони розраховані на тих, хто здатний сприймати світло. [3]



Рисунок 1.6 Розумні окуляри для погано зрячих людей [3]

7) Тифлофлешплеєр/GPS-навігатор

Тифлофлешплеєр/GPS-навігатор для незрячих з підтримкою формату DAISY, апаратура для запису та відтворення аудіо- та відеоматеріалу для незрячих.

Пристрій поєднує в собі просте читання книг і відтворення мультимедіа з технологією навігації. В даний час люди в усьому світі споживають інформацію під час подорожей, але для незрячих або слабозорих людей фізичні клавіші і звичний для користувача інтерфейс в дорозі набагато важливіше, коли одна рука вже зайнята тростиною або повідцем собаки-провідника.

Саме ці потреби враховані в пристрої. Пристрій дозволяє перевірити, де знаходиться незрячий. в режимі «Орієнтування» достатньо натиснути клавішу «Де я».

Натисканням однієї кнопки Trek пристрій повідомить: поточну найближчу адресу в якому напрямку рухається незрячий; опис наступного перехрестя; подальші вказівки, якщо користувач йде за певним маршрутом.

При встановленні голосової геомітки можна отримувати вказівки при кожному повороті до цієї конкретної точки. Щоб пройти за конкретною адресою, клавіатуру Пристрою можна використовувати як клавіатуру телефону, для вводу відповідного тексту.

Пристрій надає окремі інструкції для пішоходів і транспортних засобів: пішохідні маршрути гарантують, що користувач не отримає вказівки рухатися в бік великих автомагістралей.

В автомобільних маршрутах надається інформація менш деталізована для спрощення допомоги зрячому водієві при навігації. Адміністратори або інструктори з орієнтування та мобільності можуть завантажувати GPX-файли, що містять точне місце розташування будівель, і додавати їх в список геоміток пристрою.



Рис. 1.7 Тифлофлешплеєр/GPS-навігатор для незрячих з підтримкою формату DAISY[4]

Пристрій використовує дані карт TomTom, які будуть регулярно оновлюватися.

Формати аудіокниг, що підтримуються: DAISY 2, 2.02, NISO Z39.86 2002/2005, NIMAS 1.1, незахищений EPUB 2, LGK.

Аудіоформати, що підтримуються: AMR-WB +, mp3, MPEG2, mp4, m4a, m4v, OGG Vorbis, Wav P.C.M, Speex.

Типи файлів, що підтримуються: 3gp, bra, brf, docx, fb2, html, m4a, m4v, mp2, mp3, mp4, mpg, OGG, rtf, SES3, txt, wav, xml, LKF.

Audible: Підтримується розширений формат (.aax).
Синтезатор мови: Acapela. [4]

РОЗДІЛ 2

ПОСТАНОВКА ЗАДАЧІ

2.1 Мета та задачі

Метою даного проекту є створення інтелектуальної системи акустично-вібраційного перетворювача для оцінки розташування об'єктів. Для досягнення такої комплексної мети та розробки системи були визначені задачі, які потрібно реалізувати:

- Огляд існуючих досліджень за темою орієнтування незрячих в просторі
- Огляд існуючих досліджень створених пристроїв для пересування незрячих в суспільстві.
- Моделювання системи: проектування бази даних, створення алгоритмів, проектування інтерфейсу
- Створення системи
- Тестування
- Введення в користування

2.2 Вибір засобів реалізації

Для реалізації обраного пристрою потрібно:

1) Arduino Pro Mini - це мікроконтролер на базі ATmega328. Він включає: 14 цифрових входів / виходів (6 з яких можна використовувати як ШІМ-виходи), 8 аналогових входів, кварцовий резонатор, кнопку скидання та контактні колодки для паяльних з'єднувачів. Шестиконтактний роз'єм можна використовувати для живлення та взаємодії з платою через USB через адаптер FTDI або Sparkfun. [5]

2) Сервопривод SG-90 (стежить привід) - привід з управлінням через негативний зворотний зв'язок, що дозволяє точно керувати параметрами руху. [6]

3) Ультразвуковий датчик відстані HC-SR04 - призначений для вимірювання відстаней від 2 до 400 см, з межею точності до 3 мм. Модуль включає ультразвуковий передавач, приймач і блок управління. На показання датчиків практично не впливає сонячне випромінювання та електромагнітний шум. На передній панелі HC-SR04 розташовані два ультразвукових датчика, перший з написом T (Передавач) - передавач ультразвукових хвиль (TCT40-16T), а другий з написом R (Receive) - приймач відбитих ультразвукових хвиль (TCT40-16R), центр - кварцовий генератор 27 МГц. [7]

4) Провід для макетування і пайки.

5) Літровий акумулятор 3.7 V 1300mAh літій-полімерний акумулятор

6) Перетворювачі частоти VFD-EL призначені для вирішення простих проблем за допомогою приводу змінного струму. Наприклад, такі як контроль швидкості насосів і вентиляторів малої потужності, стрічкові конвеєри, ротатори, невеликі обробні верстати. [8]

7) TP4056 + захист подвійну функцію 5V 1A Micro USB 18650 літієва батарея зарядний пристрій (Модуль)

8) Паяльна станція - це багатофункціональний настільний паяльник, спеціально призначений для використання в електроніці та електротехніці. Дозволяє паяти чутливі електронні компоненти з максимальним дотриманням усіх встановлених для них технічних регламентів за температурою та тривалістю пайки, рівномірністю та швидкістю нагрівання, розміром нагрівальної зони. [9]

РОЗДІЛ 3

ПРОЕКТУВАННЯ АКУСТИЧНО-ВІБРАЦІЙНОГО ПЕРЕТВОРЮВАЧА

3.1 Структура проекту

Суть даного проекту полягає в простому виготовленні пристрою, який допоможе людям з обмеженими можливостями орієнтуватися в просторі використовуючи рукавичку, яка визначає відстань до предметів за коштами ультразвукових хвиль і перетворює їх в механічні тактильні сигнали імпульси передаються за допомогою серво двигунів на руку.



Рис. 3.1 Вигляд приладу[9]

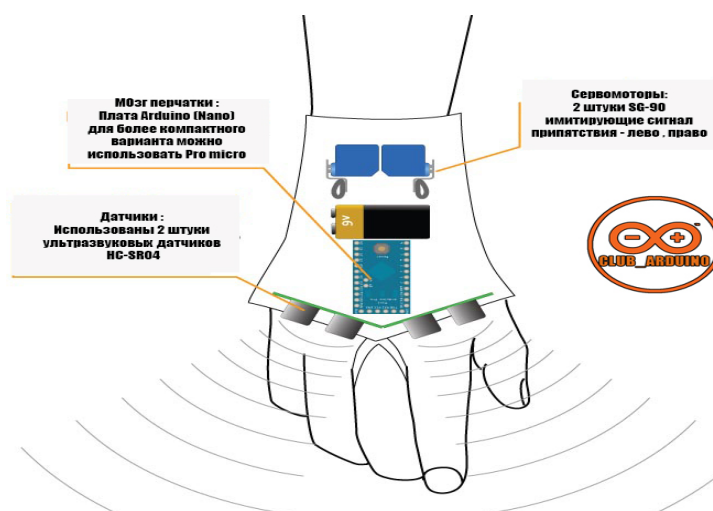


Рис .3.2 Зображення основних компонентів для збірки[9]

Прототип рукавички для людей з обмеженими можливостями:

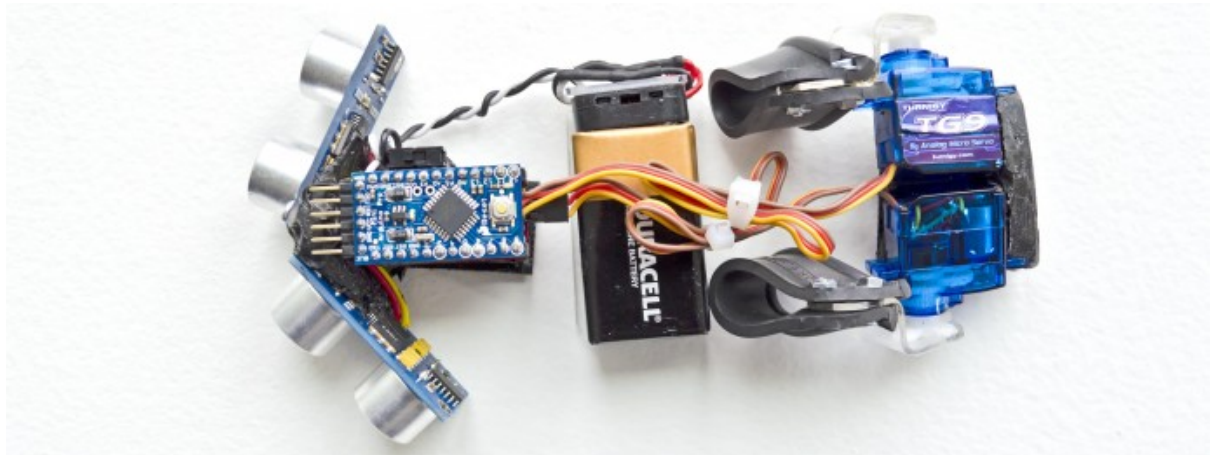


Рис. 3.3 Рукавичка для людей з обмеженими можливостями[9]

Цей прототип компактний, а також простий у монтажі. Для вдосконаленої моделі рекомендую використовувати більш надійні сервоприводи та акумулятор замість звичайної крони.

Для людей із вадами зору така рукавичка зможе доповнити орієнтацію в просторі. Конструкція цієї рукавички розроблена таким чином, щоб поєднати простоту конструкції та компактність, а також зберегти її повну функціональність. З використовуваних матеріалів - неопрен для основи самої рукавички, а також гумові вставки, які закріплені на кінцях приводів для передачі тактильного сигналу.

І так сама рукавичка одягається через великий палець і далі одним оборотом закріплюється на зап'ясті, за допомогою липучки. Таким чином рукавичку легко одягати і знімати.

3.2 Створення схеми

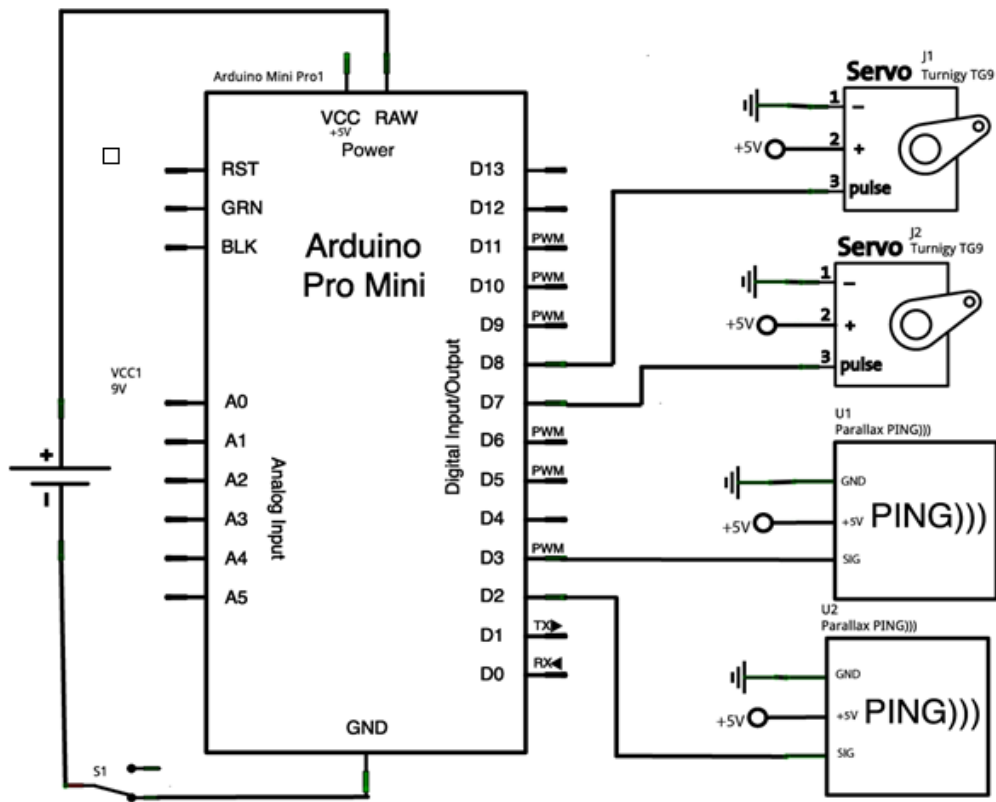


Рис.3.4 Схема проектування пристрою акустично - вібраційного перетворювача

Для розробки схеми потрібні такі складові:

- 1) Arduino PRO mini;
- 2) 2 штуки Сервоприводи SG-90 (з перехідними важелями);
- 3) 2 штуки Ультразвукові датчики SR-04;
- 4) Провід для макетування і пайки;
- 5) Акумулятор;
- 6) Перетворювач з 3.7В на 5В;
- 7) Зарядний пристрій для акб. [10]

Arduino та принципи роботи з ним

«Arduino - це платформа обчислювальної техніки, основними компонентами якої є плата вводу / виводу та середовище розробки Processing / Wiring. Arduino можна використовувати для створення самостійних інтерактивних об'єктів, а також для підключення до програмного забезпечення, що працює на вашому комп'ютері (наприклад: Adobe Flash, Processing, Max / MSP, Pure Data, SuperCollider).



Рис.3.5 Плата вводу/виводу Arduino [12]

Arduino розробляє роботи, системи спостереження та безпеки, аналоги систем розумного дому, гірлянди з «біжучими вогнями», цифровий кодований замок, пульт дистанційного керування побутовою технікою, систему автоматичного поливу квітів тощо. Повірте, це не складно, якщо ви використовуєте Arduino.

Arduino складається з мікроконтролера Atmel, а також елементів для програмування та інтеграції з іншими пристроями. Багато плат мають лінійний регулятор напруги + 5 В або + 3,3 В. Тактова частота виконується на 84, 16 або 8 МГц за допомогою кварцового резонатора. Завантажувач завантажується в мікроконтролер, тому зовнішній програміст не потрібен.

Arduino через контактні роз'єми. Інтегроване середовище розробки Arduino - це багатоплатформна програма Java, яка включає редактор коду,

компілятор та модуль передачі мікропрограмного забезпечення. Середовище розробки засноване на мові програмування Processing і розроблене для програмування новачками, незнайомими з розробкою програмного забезпечення. Мова програмування схожа на мову Wiring. Строго кажучи, це C ++, доповнений деякими бібліотеками. Програми обробляються за допомогою препроцесора, а потім компілюються за допомогою AVR-GCC.

Arduino Pro Mini - це мікроконтролер на базі ATmega328. Він включає: 14 цифрових входів / виходів (6 з яких можна використовувати як ШІМ-виходи), 8 аналогових входів, кварцовий резонатор, кнопку скидання та контактні колодки для паяльних з'єднувачів. Шестиконтактний роз'єм можна використовувати для живлення та взаємодії з платою через USB через адаптер FTDI або Sparkfun.

Arduino Pro Mini призначений для напівстаціонарні монтажу в різне устаткування або установки. Плата спеціально поставляється без впаяних роз'ємів, що дозволяє користувачеві впаявати дроти або використовувати необхідні типи роз'ємів на свій розсуд. По розташуванню висновків Arduino Mini Pro сумісний Arduino Mini.

Існує дві версії Pro Mini: одна працює від 3.3В при частоті 8 МГц, інша - від 5В при 16 МГц.

Arduino Pro Mini розроблений і виготовлений фірмою SparkFun Electronics.»[11]

Переваги:

- відкриті схеми устаткування і специфікації. Arduino виконаний на популярних мікропроцесорах Amtel і ATMEGA. Кваліфіковані фахівці можуть спроектувати на основі наявних схем власний варіант модуля для певних завдань.

- відкритий код програми. Кодування програми може розширюватися на платформі C ++.

- проста і зручна середовище програмування. Оболонка програми є легкою в застосуванні для початківців програмістів, однак має достатню

гнучкість для роботи професіоналів. Вона найзручніша для середовища навчання студентів, яким легко буде розібратися в роботі цієї платформи.

- програмування, підключення та харчування виконується одним USB-кабелем, або кабелем, що має адаптер на мікросхемі.

- можливість функціонування на різних видах систем. Програмне забезпечення успішно функціонує на Лінукс, Макінтош, Ява і інших системах, так як має відкритий код. Однак найбільш популярною системою для Ардуіно стала система Windows.

- прийнятна ціна. У великих містах Arduino можна придбати за ціною менше 100 грн. Це ціна за готове закінчений пристрій, що не вимагає допоміжного обладнання, дорогих програматорів, платних програм.

Недоліки:

- занадто убога оболонка програми. Це дуже незручний редактор коду.
- для програмування доведеться переходити на більш зручний редактор, але кодовий редактор Arduino IDE все одно необхідно залишати відкритим.
- мала частота мікропроцесора. Немає можливості її змінити.
- малий обсяг пам'яті для збереження програм.

«Завантажте скетч в Ардуіно. Скетч - додаток, написаний в середовищі Arduino. Він створюється в текстовому редакторі, який має різні інструменти: вирізати / вставити, пошук / заміну тексту. Коли проект зберігається або експортується в області повідомлень, записуються пояснення або помилки, що виникли під час компіляції.

У текстовому полі (консолі) відображається 40 повідомлень Arduino, які відображають повні повідомлення про помилки та іншу інформацію. Кнопки на панелі інструментів дозволяють перевіряти та записувати програму, створювати, відкривати та зберігати написані скетчі та відкривати моніторинг послідовної шини.

Для того, щоб завантажити скетч, необхідно задати параметри в меню Tools > Board і Tools > Serial Port. В ОС Mac послідовний порт має

позначення `dev / tty.usbserial-1B1` (для плати USB) або `/dev/tty.USA19QW1b1P1.1` (для плати послідовної шини, яка підключається через адаптер Keyspan USB-to-Serial). В ОС Windows порти мають назви: COM1 або COM2 (для плати послідовної шини) або COM4, COM5, COM7 і вище (для плати USB).

Виявлення порту USB відбувається в полі Послідовною шиною USB Диспетчера пристроїв Windows. В ОС Linux порти можуть називатися, як `/dev / ttyUSB0`, `/dev / ttyUSB1`. Після того, як був вибраний порт і платформа вибору порту потрібно натиснути кнопку завантаження на панелі інструментів або вибрати пункт меню `File> Upload to I / O Board`.



Рис. 3.6 Arduino Pro Mini[12]

Платформи Arduino сучасних видань автоматично завантажуються перед завантаженням. На платформах минулого випуску потрібно кожного разу натискати кнопку перезавантаження. Більшість плат обладнані світлодіодами RX і TX, які будуть світитися під час роботи. Після завантаження з'явиться повідомлення, коли завантаження закінчиться або сталася помилка.

Для того, щоб завантажити скетч необхідно використовувати Завантажувач (Bootloader) Arduino, це невелика програма, що завантажується в мікроконтролер на платі. З її допомогою можна завантажувати проп'є гра ний код без застосування додаткових пристроїв. Завантажувач (Bootloader)

працює при перезавантаженні платформи та при завантаженні будь-якого скетчу на мікроконтролер. При роботі завантажувача (Bootloader) миготить світлодіод.» [12]

3.3 Написання програмної частини

```
«#include
const int MaxSensors = 2;           // Якість сенсорів.
const int ServoPins[MaxSensors] = {7, 8};    // Піни для підключення серво
const int RangingPins[MaxSensors] = {3, 2};  // Піни для підключення датчиків SR
const int ReadingsPerSensor = 5;             // значення позиціонування.
const int TimePerDegree = 9;                 //
const int MinimumTurnDistance = 3;           // Захист від брязкоту, мінімальне значення для повороту на 1*
```

Підключення пристрою та його захист.

```
// Initialize the servo location and move them through a full range of motion so we know they work.
```

```
for (int i = 0; i < MaxSensors; i++){
    ServoList[i].attach(ServoPins[i]);
    delay(10);
    ServoList[i].write(ServoClose[i]);
    delay(500);
    ServoList[i].write(ServoFar[i]);
```

```

    delay(500);
    ServoList[i].detach();
}
delay(100);

```

Ініціалізування розташування сервоприводу та переміщення їх через повний діапазон руху, щоб знати, що вони працюють

```

// Initialize the sensor and tell it to send out a ping.
pinMode(RangingPins[sensorNumber], OUTPUT);
digitalWrite(RangingPins[sensorNumber], LOW);
delayMicroseconds(2);
digitalWrite(RangingPins[sensorNumber], HIGH);
delayMicroseconds(5);
digitalWrite(RangingPins[sensorNumber], LOW);

```

Ініціалізування датчика і надсилання пінгу.

```

// Read the time in milliseconds until the value comes back.
pinMode(RangingPins[sensorNumber], INPUT);
duration = pulseIn(RangingPins[sensorNumber], HIGH);

```

Час у мілісекундах до повернення значення.

// Trim the data into minimums and maximums and map it to the 0-100 output range.

```

duration = constrain(duration, SenorClose, SensorFar);
out = map(duration, SenorClose, SensorFar, 0, 100);

```

```
    return out;  
}  
//
```

Розділення даних на мінімуми та максимуми та відобразити їх у діапазоні виходів 0-100.»[12]

Повний програмний код знаходиться в додатку А.

ВИСНОВКИ

У даній дипломній роботі розглянуто прилад для орієнтування сліпих, що визначає відстань від людини до навколишніх предметів за допомогою вібрацій.

Такий пристрій допоможе незрячій людині, оскільки з його допомогою слабозорий зможе орієнтуватися на місцевості. Для сліпого підійде пристрій, який буде визначати предмети на відстані не більше 1,5 м. При наближенні незрячого до об'єктів, прилад видає акустично-вібраційні сигнали з періодичністю. У процесі скорочення відстані вібрація частішає при збільшенні відстані вібрація затухає. Якщо в області дії не знаходиться предмет, то сигнали не генеруються. Такий підхід дає повне орієнтування на місцевості та в приміщенні. Людина залежно від реакції приладу, приймає відповідні рішення по переміщенню.

Запропонований прилад буде кріпитись до руки незрячого в якості рукавиці. Витягнутою рукою сліпий виконує рухи, які дозволяють визначати предмети, розміщені на певній відстані. [13]

Управління процесом виконується мікроконтролером Arduino Pro Mini на базі ATmega328 по заданій управляючій програмі. Завдяки програмуванню можна оптимізувати роботу приладу, не вимагаючи при цьому великих енерговитрат.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Інноваційні розробки університетів і наукових установ МОН України. Т. 2 / Колектив авторів за загальною редакцією М. Стріхи та М. Ільченка. К.: ТОВ «Мірал», 2018 288 с. <https://mon.gov.ua/storage/app/media/news/%D0%9D%D0%BE%D0%B2%D0%B8%D0%BD%D0%B8/2019/01/28/innovations2018-vse.pdf> © 2016–2020.
- 2) Лазаренко В.І., Лазаренко В.П., Продайко В.М., Сурякова М.В. Психологія (Опорний конспект лекцій): Електронний навчальний посібник. 2-е видання. – Дніпропетровськ: ДНУ, 2015.– 95 с.
- 3) 10 кращих винаходів, які сліпих роблять зрячими <https://www.crimea.kp.ru/daily/25770/2755049/> /© 2019–2020.
- 4) <https://www.dzo.com.ua/tenders/4401275/complaints> = © 2014 - 2020
- 5) <https://doc.arduino.ua/ru/hardware/ProMini> ВСЕ ПРАВА ЗАЩИЩЕНЫ "ARDUINO" COPYRIGHT © 2020
- 6) <https://habr.com/ru/post/400631> © 2006 – 2020 «ТМ»
- 7) https://wiki.tntu.edu.ua_Arduino_HC_SR04 Creative Commons Attribution/Share-Alike © 2014-2020
- 8) <http://www.rts.ua/rus/catshop/736/0/23480/delta-electronics/> 2013 © АО "Системы реального времени - Украина"
- 9) <https://www.smt-prof.com.ua/BD%D1%8B%D0%B9-%D1%80%D0%BE%D0%B1%D0%BE%D1%82/> © 2020 SMT PROF · Designed by Themes & Co ·
- 10) <http://www.electronica52.in.ua/proekty-arduino/perchatka-dlya-orientirovaniya-v-prostranstve-na-2h-ultrazvukovyh-datchikah-i-arduino-> Рукавичка для орієнтування в просторі на 2х ультразвукових датчиках HC-SR04 і ARDUINO © 2010–2020.
- 11) <https://amperka.ru/page/what-is-arduino> © 2010–2020. ООО «Амперка»

12) https://gb.mistoboyarka.gov.ua/files/project/1632/documents/15120646387546_1512063065317996.pdf Основні елементи Arduino, макетний стенд Arduino і редактор програм ©2020

13) ПРИЛАД ДЛЯ ОРІЄНТУВАННЯ СЛІПИХ НА МІСЦЕВОСТІ
<https://core.ac.uk/download/pdf/14042139.pdf>

Додаток А

Програмний код присторою

```

#include

const int MaxSensors = 2;           // Якість сенсорів.

const int ServoPins[MaxSensors] = {7, 8};    // Піни для підключення серво

const int RangingPins[MaxSensors] = {3, 2};  // Піни для підключення датчиків SR

const int ReadingsPerSensor = 5;           // значення позиціонування.

const int TimePerDegree = 9;              //

const int MinimumTurnDistance = 3;         // Захист від брязкоту, мінімальне значення для повороту на 1*

// Variables

Servo ServoList[MaxSensors];

int sensorReadings[MaxSensors][ReadingsPerSensor]; // Утримання показань кожного датчика

int calculatedSensorReadings[MaxSensors];           // Визначення відстані для кожного датчика

int latestReading = 0;                             // Поточна позиція в масиві для останнього читання.

int servoLocations[MaxSensors];                    // Позиціонувати серво

int SensorClose = 500;                             // Фіксація мінімальної відстані - відгук.)

int SensorFar = 14000;                             // Дальня відстань. Відгук звукових серво PING sensor. (Пройдення звукової хвилі.)

int ServoClose[MaxSensors] = {0, 160};             // Кут повороту першої серви.

```



```

int ServoFar[MaxSensors] = {70,110};           // Кут пов
ороту серви - можна коригувати.

void setup() {

    //Serial.begin(115200);                     // Для відображення
даних і настройки прибрати // .

    //Serial.println("Begin...");

    // Initialize the servo location and move them through a full
range of motion so we know they work.
    for (int i = 0; i < MaxSensors; i++){
        ServoList[i].attach(ServoPins[i]);
        delay(10);
        ServoList[i].write(ServoClose[i]);
        delay(500);
        ServoList[i].write(ServoFar[i]);
        delay(500);
        ServoList[i].detach();
    }
    delay(100);
}

void loop(){
    int i, j, oldLocation;
    unsigned long delayTime;

```

```

// Loop through each range sensor

for (i = 0; i < MaxSensors; i++){    // Get the current sensor's range.
    sensorReadings[i][latestReading] = getDistance(i);
    // Figure out an averaged/smoothed readings based on this and past data.
    calculatedSensorReadings[i] = calculateNewDistance(i);
    // Set the servo to the correct angle.
    oldLocation = servoLocations[i];
    servoLocations[i] = map(calculatedSensorReadings[i], 0, 100, ServoClose[i], ServoFar[i]);
    if (latestReading >= ReadingsPerSensor-1){
        // Don't do anything until we have enough data to trend.

        if (abs(servoLocations[i]-oldLocation) >= MinimumTurnDistance){
            // Only try to turn it if we have somewhere to go.

            ServoList[i].attach(ServoPins[i]);

            delay(10);

            ServoList[i].write(servoLocations[i]);

            delayTime = (TimePerDegree * (abs(servoLocations[i]-oldLocation))+20);
            // Set a delay for the next reading so motor noise doesn't interfere with sensor readings.

            if (abs(delayTime)>500){ // If it can't do it in this amount of time
                // It's based on how far it has to turn to keep the delay to a minimum, response time at a maximum.

                delayTime=500;           // we'll get it next time. Keep it responsive.
            }

            delay(delayTime);

            ServoList[i].detach();

        } else {
            // Otherwise if the reading hasn't changed enough write the old value to

            ServoList[i].attach(ServoPins[i]);           // the servo so that it will hold in place if it's applying pressure.
        }
    }
}

```

```

        delay(10);
        ServoList[i].write(oldLocation);
        delay(50);
        ServoList[i].detach();
        servoLocations[i]=oldLocation;
    }
}

    delay(20); // Added to fix left sensor misbehavior reported
by Rob.

}

    latestReading++; // Increment the reading counter so we know
where we're at.

    if (latestReading >= ReadingsPerSensor){ // Make sure we don
't record more readings than we have space to hold.

        latestReading = ReadingsPerSensor-1;

        // Pop the oldest reading off the list.

        for (i = 0; i < MaxSensors; i++){

            for (j=0; j < ReadingsPerSensor-1; j++){
                sensorRe
adings[i][j] = sensorReadings[i][j+1];
            } } } // f
unction: calculateNewDistace(sensorNumber: Which sensor's data
to process): Calculated distance in 0-100 range. // Apply some
averaging and smoothing to the recorded distance readings // to
take care of noisy data. int calculateNewDistace(int sensorNumb
er){    int output = SensorFar;                // Default
value is the furthest distance.    float weightingFactor = 0.5;
// How fast the reading's importance tapers off in time. (1= no
taper, 0 = divide by zero error.)    float flickerFactor = 30;
// When the change is greater than this, ignore it unless its t
wo in a row. (It's probably noise.)    if (latestReading >= Read

```

```

ingsPerSensor-1) { // Only do this if we have a full set of re
adings to sample.

    int total = 0; // Average them
with a weighting.

    float currentWeight = 1; // New readings
count more than older readings.

    float percentagePossible = 0;

    boolean flickered = false;

    for (int i=ReadingsPerSensor-1; i >=0 ;i--){ // Check for
flicker (This reduces jitter with something right on the thresh
old.)

        flickered = false;

        if (i==ReadingsPerSensor-1){

            if ((abs(sensorReadings[sensorNumber][i])-abs(sensorRea
dings[sensorNumber][i-1]) > flickerFactor) &&

                (abs(sensorReadings[sensorNumber][i-1])-abs(sensorRe
adings[sensorNumber][i-2]) > flickerFactor)){

                flickered = true;

            }

        }

        if (flickered==false){

            total += (sensorReadings[sensorNumber][i] * currentWeig
ht);

            percentagePossible += currentWeight;

            currentWeight *= weightingFactor;

        }

    }

    output = total / percentagePossible;

```

```

    }

    return output;
}

// function: getDistance

// Take a sensor number (not pin number) and returns an int in
// the 0-100 range

// 0 = closest, 100= furthest. (It's a percentage of the distance
// that the software

//

// Note: Function is designed to be generic so that it can be swapped
// out for

//      different kinds of ranging sensors.

//      This version of the function is made for Parallax PING
//      ))) sensors

//      For more info see http://arduino.cc/en/Tutorial/Ping
//
//      and http://www.parallax.com/tabid/768/ProductID/92/Default.aspx

int getDistance(int sensorNumber){

    long duration;    // How long it takes a sonic pulse to reflect
    back.

    int out;          // The value we send back from the function

    // Initialize the sensor and tell it to send out a ping.
    pinMode(RangingPins[sensorNumber], OUTPUT);
    digitalWrite(RangingPins[sensorNumber], LOW);
    delayMicroseconds(2);
    digitalWrite(RangingPins[sensorNumber], HIGH);
    delayMicroseconds(5);

```

```
digitalWrite(RangingPins[sensorNumber], LOW);

// Read the time in milliseconds until the value comes back.
pinMode(RangingPins[sensorNumber], INPUT);
duration = pulseIn(RangingPins[sensorNumber], HIGH);

// Trim the data into minimums and maximums and map it to the
0-100 output range.
duration = constrain(duration, SenorClose, SensorFar);
out = map(duration, SenorClose, SensorFar, 0, 100);
return out;
}»[12]
```